

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PUCP

**VISUALIZACIÓN 3D DE GRANDES CANTIDADES DE DATOS 3D
PARA LA PREVENCIÓN FRENTE A DESASTRES NATURALES: UNA
REVISIÓN DE LA LITERATURA
TRABAJO DE INVESTIGACIÓN PARA LA OBTENCIÓN DEL
GRADO DE BACHILLER EN CIENCIAS CON MENCIÓN EN
INGENIERÍA INFORMÁTICA**

AUTOR

GUILLÉN ZAPATA, HANDRY JAMES

ASESOR:

SIPIRAN MENDOZA, IVÁN ANSELMO

Lima, Junio, 2020

Resumen

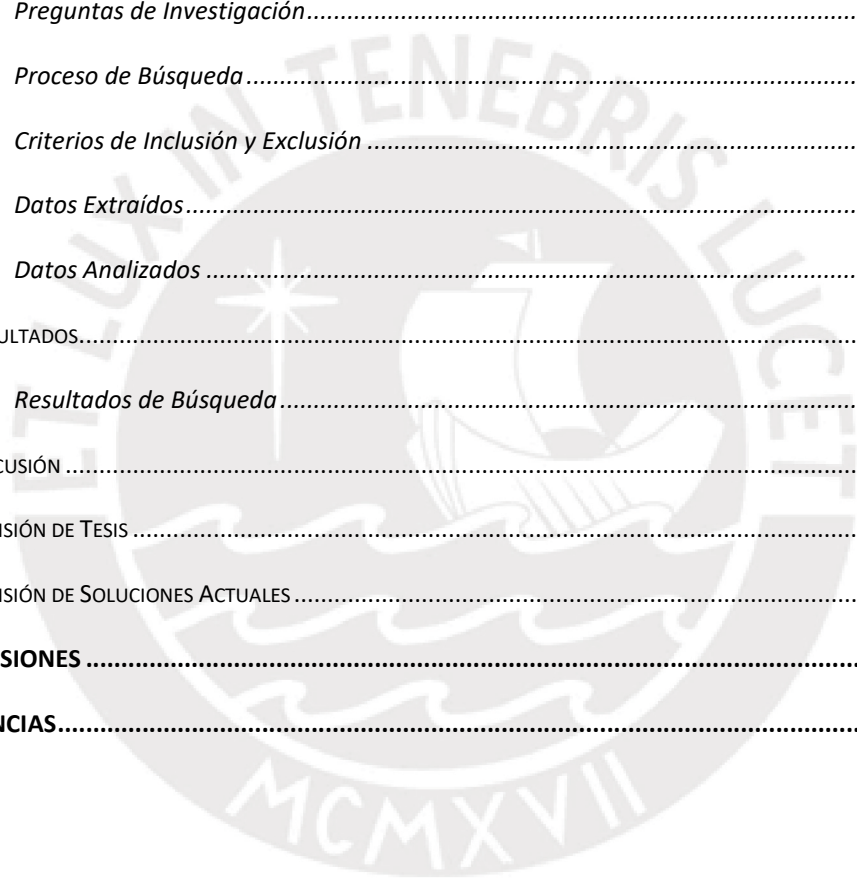
La visualización de datos 3D es un aspecto muy importante para varios campos de trabajo como la arquitectura, minería, videojuegos, diseño gráfico, geografía, etc. En especial en este último, la geografía, el cual a diario necesita visualizar información topográfica para hacer un estudio previo de los terrenos sin la necesidad de estar presente.

En Perú, el uso información 3D por parte de los geógrafos para analizar terrenos a detalle, se ha vuelto una actividad de mucha importancia, debido a que el país es muy propenso a sufrir de fenómenos naturales como el fenómeno del Niño. Esta información topográfica suele ser de superficies de un gran tamaño que pueden llegar hasta un área de 7000 hectáreas, por lo que se necesita de procesar una inmensa cantidad de información 3D.

Es por esto que el presente trabajo de investigación se centra en revisar la literatura para lograr una visualización de grandes cantidades de datos 3D. En primer lugar, se revisará cómo es que se deben guardar y organizar los datos 3D para que puedan ser fácilmente extraídos por el visualizador y qué tipo de estructura de datos es la mejor para este tipo de trabajos. En segundo lugar, se revisarán los métodos que existen actualmente para poder renderizar los datos 3D de forma fluida, con el objetivo de lograr una visualización interactiva del usuario sin exigir tantos recursos. Por último, se presentarán las conclusiones de los dos puntos mencionados anteriormente y se explicarán cuáles son los mejores métodos para realizar el proyecto de visualización de grandes cantidades de datos 3D.

Contenido

RESUMEN	2
CONTENIDO	3
ÍNDICE DE TABLAS	4
1 INTRODUCCIÓN	5
2 MÉTODO	6
2.1 REVISIÓN SISTEMÁTICA	6
2.1.1 Preguntas de Investigación.....	6
2.1.2 Proceso de Búsqueda.....	6
2.1.3 Criterios de Inclusión y Exclusión	6
2.1.4 Datos Extraídos.....	7
2.1.5 Datos Analizados	7
2.2 RESULTADOS.....	7
2.2.1 Resultados de Búsqueda.....	7
2.3 DISCUSIÓN	12
2.4 REVISIÓN DE TESIS	15
2.5 REVISIÓN DE SOLUCIONES ACTUALES	16
3 CONCLUSIONES	17
4 REFERENCIAS	18



Índice de Figura

Ilustración 2.1 Imagen del sistema AHN Viewer y sus funcionalidades..... 16

Índice de Tablas

Tabla 2.1 Cantidad de artículos encontrado en cada repositorio..... 7

Tabla 2.2 Artículos finales que se usarán de referencia en el proyecto..... 8



1 Introducción

Perú es un país propenso a sufrir de varios fenómenos naturales, los cuales suelen causar mucho daño, debido a la incertidumbre con las que estos suelen suceder. Un ejemplo claro es lo sucedido el 2017, en ese año se dio uno de los Fenómenos del Niño que más estragos causó. El país claramente no estaba preparado del todo para afrontar a un fenómeno natural de tales magnitudes; por lo que una de las causas principales de los grandes daños ocasionados fue la no resiliencia de las entidades encargadas para la prevención frente a estos desastres. (Venkateswaran et al.,2017)

Según INDECI, la falta de datos topográficos ha sido una de las principales razones por las cuales las entidades no han podido prevenir de mejor manera los daños causados por los desastres, ya que a partir de la visualización de todos estos terrenos se pueden realizar estudios hidrográficos más exactos y definir tareas de descolmatación o construcción de diques. (INDECI, 2019)

Usualmente los terrenos que se quieren visualizar por las entidades son de grandes tamaños, ya que involucran pueblos o hasta ciudades. Lograr la visualización de estos datos topográficos requiere un buen manejo y procesamiento de los datos por la gran cantidad que estos representan. Además, lo que se busca es que esta visualización se pueda realizar en la mayoría de computadoras, considerando las limitaciones que esta tengan con respecto a su unidad de procesamiento gráfico (GPU). (Schuetz & Wimmer, 2018)

Es por ello que el gran problema que se tiene es el de la visualización de estas zonas vulnerables. Esto involucra 2 aspectos importantes, el primero involucra el almacenamiento y extracción de datos 3D; al tratarse de una gran cantidad de datos 3D se necesita de una estructura de datos que pueda almacenar toda esta información y, también, en la cual sea posible poder extraer la información suficiente y necesaria. Esto último es indispensable, ya que con esa función se asegura que la información que será enviada no ocupe toda la memoria del GPU (la cual es limitada). El segundo aspecto involucra los algoritmos de renderizado, estos permitirán mostrar en pantalla todos los datos 3D extraídos de la estructura de datos; por ello, los algoritmos deben ser los adecuados para cumplir la visualización de entornos montañosos, zonas rocosas, valles, campos, ciudades, etc., con el objetivo de ofrecer un entorno realista. (Schütz, 2016)

2 Método

2.1 Revisión Sistemática

2.1.1 Preguntas de Investigación

El problema de visualización de grandes cantidades de datos 3D toma como objetivo principal el poder almacenar y procesar toda la información necesaria para ser visualizada. Por lo tanto, el primer punto que debe ser abordado es el de la selección de la estructura de datos que involucran el almacenamiento y extracción de los datos 3D. Y como segundo punto, se tiene el establecimiento de los algoritmos de renderizado e iluminación, los cuales involucran la visualización interactiva de los datos extraídos a través de las estructuras de datos.

Como complemento de información, se intentan responder las siguientes preguntas para la problemática planteada:

- ¿Qué tipo de estructura de datos es la adecuada para poder almacenar y extraer la gran cantidad de nube de puntos 3D de los terrenos afectados por los desastres?
- ¿Qué tipos de renderizados se deben implementar para lograr una visualización interactiva de los terrenos afectados por los desastres?

2.1.2 Proceso de Búsqueda

El proceso de búsqueda se realizará en las bases de datos más importantes sobre artículos científicos de ingeniería y ciencias de la computación, esto con el objetivo de obtener una amplia variedad de investigaciones relacionadas al problema que puedan aportar a nuestro estado del arte. Finalmente, con esto se obtendrán ideas más compactas que puedan responder las preguntas de investigación.

2.1.3 Criterios de Inclusión y Exclusión

Tomando en cuenta la primera pregunta de investigación, se propuso la siguiente cadena de búsqueda:

TITLE-ABS-KEY (visualization AND “point cloud” AND “data structure”)

De la relación de artículos que encontremos con la cadena de búsqueda, excluirémos las que traten sobre estructuras de datos complejas que hagan uso de inteligencia artificial, ya que estos escapan del alcance de este proyecto de tesis.

Por otro lado, tomando en cuenta la segunda pregunta de investigación se propuso la siguiente cadena de búsqueda:

TITLE-ABS-KEY (algorithm AND (render OR rendering) AND “large point cloud”)

De la relación de artículos que encontremos con las cadenas de búsquedas, excluirémos las que trate sobre algoritmos de renderizado únicamente diseñados para entornos de realidad virtual y, también, algoritmos que no han arrojado buenos resultados en su implementación.

2.1.4 Datos Extraídos

Los datos que se están extrayendo de cada artículo, principalmente corresponde a la parte de implementación de algoritmos (estructura de datos y renderizado) y de resultados, con el objetivo de corroborar la efectividad de los algoritmos estudiados en cada artículo y elegir el más adecuado para este tipo de trabajos.

2.1.5 Datos Analizados

Los artículos analizados tendrán años de publicación entre el 2004 y el 2019, debido a que el problema de visualizar grandes cantidades de datos 3D ha sido abordado desde hace tiempo y algunos conceptos antiguos siguen vigentes al día de hoy. Para la selección de artículos, no se le dará una preferencia a determinado países; sin embargo, solo se le dará importancia, en su mayoría, a artículos redactados en inglés y español de reconocidas editoriales científicas, congresos de ciencias y eventos de alta reputación donde se publiquen artículos científicos relacionados al tema.

2.2 Resultados

2.2.1 Resultados de Búsqueda

A partir de las cadenas de búsqueda creadas, se inició la búsqueda de artículos científicos en las plataformas más conocidas para este rubro.

Tabla 2.1 Cantidad de artículos encontrado en cada repositorio

Base de datos o repositorio	Artículos encontrados con la primera cadena de búsqueda	Artículos encontrados con la segunda cadena de búsqueda
Scopus	77	33
IEEE Electronic Library	109	89
ACM	123	112

Repositorio de Tesis de Informática	2	1
-------------------------------------	---	---

Luego de identificar los artículos repetidos entre repositorios y de aplicar los criterios de inclusión y exclusión, obtenemos la siguiente lista de artículos:

Tabla 2.2 Artículos finales que se usarán de referencia en el proyecto

Nombre del artículo	Base de Datos	Información relevante del artículo	Autores
Simple Octree Solution for Multi-resolution LiDAR Processing and Visualisation	IEEE	Propone un método para visualizar grandes cantidades de puntos usando una estructura Octree simple, pero agregando un mapa de reducción hecho en Python.	Cha, G., Park, S., & Oh, T. (2019).
Out-of-core gpu-based change detection in massive 3D point clouds	Scopus	Justifica el uso del Octree como estructura de datos, desplaza al KD-Tree, por el hecho de que la construcción de este se demora mucho más para grandes cantidades de puntos.	Richter, R., Kyprianidis, J. E., & Döllner, J. (2013)
A scalable WebGL-based approach for visualizing massive 3D point clouds using semantics-dependent rendering techniques	ACM	Utiliza diferentes tipos de árboles dependiendo del conjunto de datos escogidos	Discher, S., Richter, R., & Döllner, J. (2018, June)
Real-Time Continuous Level of Detail Rendering of Point Clouds	IEEE	Los autores hablan sobre una nueva estructura de datos llamado Continuous	Schütz, M., Krösl, K., & Wimmer, M. (2019, March)

		Level Of Detail Rendering, al parecer esta es buena para ambientes con movimiento constante.	
A Terrestrial LiDAR-Based Detection of Shape Deformation for Maintenance of Bridge Structures	Scopus	Presenta un resumen de qué son realmente los sistemas de visualización	Cha, G., Park, S., & Oh, T. (2019)
Efficient Randomized Hierarchy Construction for Interactive Visualization of Large-Scale Point Clouds	IEEE	Hace uso de un MNO (octree) combinado con muestras aleatoria	Kang, L., Jiang, J., Wei, Y., & Xie, Y. (2019, June)
Classification and Integration Of Massive 3D Points Clouds In A Virtual Reality (VR) Environment	Scopus	Se usa la estructura de datos implementada en Potree; además, Eye Dome Lightning como shader de iluminación, incluyendo otros aspectos para evitar la oclusión entre puntos	Kharroubi, A., Hajji, R., Billen, R., & Poux, F. (2019)
Explorations for Real-Time Point Cloud Rendering Of Natural Scenes In Virtual Reality	IEEE	Los autores describen al Octree como estructura de datos principal. También muestran cómo se extrae los datos de la escena que se quiere renderizar	Bonatto, D., Rogge, S., Schenkel, A., Ercek, R., & Lafruit, G. (2016, December)
A Review of Data Structure and Filtering in Handling 3D Big Point Cloud Data for Building Preservation	IEEE	Hace hincapié sobre varias estructuras de datos que se pueden implementar para el manejo de grandes nubes de puntos 3D	Isa, S. M., Shukor, S. A., Rahim, N. A., Maarof, I., Yahya, Z. R., Zakaria, A., ... & Wong, R. (2018, December)

Smart Point Cloud: Definition and Remaining Challenges	Scopus	Presenta una nueva estructura de datos justificando su viabilidad	
Buildings and Terrain of Urban Area Point Cloud Segmentation based on PCL	Scopus	Presenta una estructura de datos KD-Tree para la visualización de construcciones y terrenos urbanos, pero no justifica su uso	Liu, Y., & Zhong, R. (2014, March)
Analysis of Interactive Editing Operations for Out-of-Core Point-Cloud Hierarchies	Scopus	Presenta dos variantes del Octree MNO y MWO, las compara y da su veredicto.	Scheiblauer, C., & Wimmer, M. (2013)
One billion points in the cloud – an octree for efficient processing of 3D laser scans	Scopus	Este artículo habla sobre la eficiencia en compresión de tamaño de datos que ofrece el Octree.	Elseberg, J., Borrmann, D., & Nüchter, A. (2013).
Efficient Processing of Large 3D Point Clouds	IEEE	Este artículo habla sobre el Octree y de cómo ha sido implementado para manejar una gran cantidad de puntos 3D	Elseberg, J., Borrmann, D., & Nüchter, A. (2011, October)
An Efficient Point Cloud Management Method Based on a 3D R-Tree	Scopus	Presenta una combinación entre Octree y 3D R-Trees	Gong, J., Zhu, Q., Zhong, R., Zhang, Y., & Xie, X. (2012)
Out-of-Core Real-Time Visualization of Massive 3D Point Clouds	ACM	Este artículo presenta un camino de cómo renderizar una gran cantidad de nube de puntos usando una estructura Octree y aplicando un algoritmo de renderizado especial	Richter, R., & Döllner, J. (2010, June).

		para este tipo de trabajos.	
Interactive Editing of Large Point Clouds	Scopus	Este artículo presenta una herramienta interactiva para editar características de grandes cantidades de puntos. Usa Octrees para la visualización de estos.	Wand, M., Berner, A., Bokeloh, M., Fleck, A., Hoffmann, M., Jenke, P., ... & Schilling, A. (2007)
Web-Based Visualisation Of On-Set Point Cloud Data	ACM	Un sistema web de visualización que toma como estructura el Octree	Evans, A., Agenjo, J., & Blat, J. (2014, November).
Visualization of point clouds built from 3D scanning in coal mines	IEEE	Un sistema de visualización que utiliza Direct 3D y normales para la iluminación.	Kot, T., Novak, P., & Babjak, J. (2016, May).
OMiCroN – Oblique Multipass Hierarchy Creation while Navigating	Scopus	Presenta un novedoso algoritmo de renderización	da Silva, V., Esperança, C., & Marroquim, R. (2019)
Point-base rendering	Scopus	Presentan las técnicas de renderizado utilizadas frecuentemente.	Sainz, M., & Pajarola, R. (2004).
Simulating the Experience of Home Environments	IEEE	Presenta el flujo de trabajo de cómo se renderiza los puntos a partir de un Octree	Ponto, K., Tredinnick, R., & Casper, G. (2017, June)
SIFT-Realistic Rendering	IEEE	Contiene información sobre la técnica de Splat Rendering	Sibbing, D., Sattler, T., Leibe, B., & Kobbelt, L. (2013, June)

Visually-Complete Aerial LiDAR Point Cloud Rendering	IEEE	Explica un tipo de renderizado utilizando otras formas aparte de puntos. También usa Splat Rendering.	Gao, Z., Nocera, L., & Neumann, U. (2012, November)
--	------	---	---

2.3 Discusión

A continuación, se procederá a resolver las preguntas planteadas en el inciso 2.1.1.

- **¿Qué tipo de estructura de datos es la adecuada para poder almacenar y extraer la gran cantidad de nube de puntos 3D de los terrenos afectados por los desastres?**

Para responder la primera pregunta, ¿Qué tipo de estructura de datos es la adecuada para poder almacenar y extraer la gran cantidad de nube de puntos 3D de los terrenos afectados por los desastres? Se identificaron los artículos relacionados a la selección de estructura de datos según las tareas que se necesitaban realizar. En primer lugar, un estudio realizado por los investigadores Elseberg, Borrman y Nüchter, describe como realizaron el procesamiento de 1000 millones de puntos a partir de una estructura Octree. Ellos encontraron a la estructura de datos Octree como la ideal para el almacenamiento de nube puntos extraídos de un scanner, debido a que el scanner solo extraía los puntos de la superficie que representaban los objetos. El hecho de que tengan los objetos representados como sus superficies en nube de puntos, benefició a que el Octree no ocupe tanto espacio y redujera de manera significativa la carga de datos; es decir, los puntos se cargaban más rápido cuando se almacenaban en Octrees y no tenían pérdidas en la precisión de localización de los puntos. (Elsebeg, Borrman, & Nüchter,2013)

Un año después, se realizó una investigación que presentaba a una estructura de datos Octree para tareas como la de almacenar diferentes nubes de puntos 3D y poder acceder a toda la información de manera rápida. Para ello, crean el Octree fuera del proceso de visualización de datos (out-of-core), esto con el objetivo de que cuando se requiera de la visualización de todos los objetos, la comunicación sea directa con el Octree y no con los archivos que guardan las nubes de puntos individualmente. Esto ahorra bastante tiempo en la carga de los objetos y, además, reduce el espacio de almacenamiento de las nubes de puntos. La investigación también recomienda no usar la estructura de datos KD-Tree para almacenar varias nubes de puntos, ya que es difícil construir este tipo de estructura con más de una

nube de puntos; además, en caso de construirse, el preprocesamiento se incrementaría considerablemente, debido a que el árbol generado tendrá una mayor cantidad de niveles y nodos. (Richter, & Döllner, 2010)

Unos años más tarde, Schütz (2016) introduce un sistema web de visualización de nube de puntos, el cual puede renderizar grandes cantidades de nube de puntos en navegadores web. Este sistema, desarrollado en WebGL, permite visualizar conjuntos de datos que contienen billones de puntos en tiempo real. Este renderizado de billones de puntos se logra gracias a la estructura jerárquica de datos que presenta, el cual denominó Potree's Octree Structure. Esta estructura es una variación de Modifiable Nested Octree (un Octree modificado), la diferencia es que en este último no se define la resolución de cada nodo por la distancia entre los puntos presentes (espaciado entre puntos). Por otro lado, el número de puntos de entrada es un factor que afecta la profundidad del Octree (el tiempo de carga se ve afectado); por ello, para reducir los tiempos de carga, el autor propuso dividir por trozos el árbol tal que cada muestra contenga una jerarquía de menor a mayor cantidad de datos, esto quiere decir que mientras se vaya recorriendo el árbol desde el primer nivel hasta el último, la resolución irá aumentando.

Luego de los resultados positivos que dio la estructura de datos implementada por Schütz, aparecieron estudios que lo tomaron como referencia. Uno de ellos es el que propone usar la estructura de datos creada por Schütz para almacenar nubes de puntos clasificados, esto quiere decir que dentro del Octree existen diferentes nodos que indican el tipo de nube de puntos según su clasificación, con el objetivo de integrarlo a un sistema de realidad virtual. (Kharroubi et al., 2019)

Ese mismo año, los investigadores Kang, Jiang, Wei y Xie toman como pieza clave la estructura que modificó Schütz (Modifiable Nested Octree), añadiendo una nueva característica la cual es realizar un submuestreo aleatorio de los puntos dentro del Octree con el objetivo de acelerar la creación del mismo. Otro punto importante a considerar es la técnica de divide y vencerás que realizan para construir el Octree con todas las nubes de puntos del conjunto de datos. Esta técnica consiste en dividir todo el conjunto de datos en bloques anidados pequeños, luego crean paralelamente los Octrees con cada bloque, y por último unen todos estos Octrees para obtener el Octree final definitivo con toda la nube de puntos. (Kang et al., 2019)

Finalmente, existen otras estructuras que no están relacionada a los Octrees o árboles en general; sin embargo, están destinados a realizar otros tipos de tareas más complejas. Un ejemplo es el realizado por Schütz, Krösl y Wimmer, los cuales crearon una estructura para realizar renderización continua con nivel de detalle de

nube de puntos, esto con el objetivo de que la visualización en los lentes de realidad virtual sea cómoda para los usuarios. Esta estructura de datos consiste en almacenar toda la nube de puntos en un arreglo simple, el arreglo está dividido en secciones las cuales indican el nivel de detalle de la nube de puntos; es decir, cuando más profundo en el arreglo se encuentre una nube de puntos, mayor será la cantidad de puntos contenida en esa sección. (Schütz, Krösl & Wimmer, 2019)

- **¿Qué tipos de renderizados se deben implementar para lograr una visualización interactiva de los terrenos afectados por los desastres?**

Revisando la literatura presente sobre los algoritmos de renderizado, no se encontraron aplicaciones específicas para la visualización de terrenos geográficos naturales, que corresponden a la mayoría de ambientes que se desean visualizar en este trabajo. Sin embargo, las técnicas de renderizado encontrados en el estado del arte dan un enfoque general de cómo se deberían procesar los datos obtenidos de la nube de puntos para lograr una visualización interactiva.

A continuación, se presentarán algunas de las técnicas de renderizado encontradas en el estado del arte.

Una publicación del año 2004 presenta dos técnicas básicas para renderizar puntos. La primera consiste en renderizar los puntos como cuadrados alineados a la pantalla, esto presenta la ventaja de tener menos costo de procesamiento cuando se envía la posición, el color y la normal del punto. La otra alternativa que se presenta es la de renderizar los puntos como mallas de triángulos; es decir, formar los objetos a partir de triángulos que son la unión de tres puntos, logrando visualizar objetos volumétricos por todas las vistas posibles. La desventaja de este método es que es más costoso en procesamiento. (Sainz & Pajarola, 2004)

Años más tarde, los investigadores Richter y Döllner propusieron un método de renderizado con el objetivo de visualizar una gran cantidad de nube de puntos teniendo en cuenta los límites de la GPU. Ellos utilizaron el espacio proyectado de un nodo de un árbol Octree (en el Octree tenían almacenado la nube de puntos), el cual era representado como una esfera. Encontraron una dificultad en representar terrenos llanos muy extensos utilizando este método; sin embargo, esto se compensaba con la rápida detección de los puntos que la cámara visualizaba en el Octree, aumentando considerablemente la interactividad del visualizador. (Richter y Döllner, 2010)

Un trabajo realizado para encontrar un mayor realismo a la visualización de los objetos, introducía un método de renderización llamado Textured Splat Rendering. Este método consiste en proyectar cada punto como pequeñas discos con información

de sus normales y radios, con esto era posible ubicar más fácil a los puntos vecinos dentro del conjunto de puntos; además, cada punto podía guardar información de la textura de la imagen que quería representar, añadiendo un aspecto fotorrealista a la visualización. (Sibbing et al., 2013)

Dentro del trabajo realizado por Schütz (2016), el cual abarcaba la creación de un Octree modificado para el almacenamiento con nivel de detalle de la nube de puntos, se encontraba el método de renderización que utilizó para la visualización de los puntos. Este método consiste en el uso de Splat rendering en alta definición, el cual se diferencia del método tradicional de Splat rendering por la forma en la cual ubica los puntos en la pantalla. El método de Splat rendering en alta definición ubica los puntos de forma continua(pegada), en vez de separarlo por cierta distancia; esto agrega a la visualización una mejor suavizado, reduce la oclusión entre puntos y evita el aliasing (bordes en forma de montañas). En su trabajo, Schütz también muestra la ventaja de usar el Eye-Dome-Lightning como método de iluminación de los objetos, ya que reduce mucho el costo de procesado por cada punto. Al no requerir el uso de la normal por cada punto, se evita el cálculo de este y solo se concentra en agregar opacidad en partes del objeto que tengan mayor profundidad con respecto a la cámara.

Además de los métodos para renderizar por computadora, se encontraron métodos para renderizado en ambientes de realidad virtual (los cuales deben tomar como prioridad la experiencia de usuario). En el año 2017, se realizó un interesante trabajo para renderizar puntos de manera interactiva en este tipo de ambientes. Este consistía en renderizar los puntos a través de un método de renderizado por marcos; es decir, el algoritmo reutiliza la información de los puntos del marco anterior para poder renderizar al objeto desde otro punto de vista en el siguiente marco. Este método logró mejorar la experiencia de visualización de entornos domésticos en realidad virtual, logrando que las personas sientan mayor comodidad al usar los lentes de realidad virtual. (Ponto et al., 2017)

2.4 Revisión de Tesis

Durante la revisión de las tesis en el repositorio de tesis PUCP utilizando las cadenas de búsquedas descritas en la sección 2.1.2, solo se encontraron dos tesis; sin embargo, uno de ellas es la única que logra complementar la información encontrada por las preguntas de investigación.

Moreno (2014) presenta en su tesis un diseño de algoritmo para renderizado de estructuras proteicas. Su trabajo consiste en renderizar proteínas a partir de un archivo PDB, el cual contiene información de sus posiciones en el espacio tridimensional de los átomos y su

conectividad. El primer punto importante que desarrolla este trabajo es el de organizar los objetos que conforman la proteína dentro de un Octree, esto con el objetivo de solo renderizar los objetos visibles por la cámara. Para esto usa el Frustum Culling (espacio volumétrico visible de la cámara). Otro punto importante es acerca del renderizado que usa, al parecer renderizar varios objetos a la vez por medio del API de OpenGL es muy costoso; por lo tanto, Moreno decidió hacer algunos ajustes al Frustum Culling. Uno de ellos fue el de reducir el espacio volumétrico para no coger objetos que solo se vean parcialmente. Además, utilizó una nueva función disponible para la versión 4.3 de OpenGL llamada `glMultiDrawElementsIndirect()`; sin embargo, la desventaja que hay al usar esta función es que todos los objetos van a utilizar la misma función de sombreado, al no haber parámetros con los cuales indicar qué sombreado usar para cada objeto.

2.5 Revisión de Soluciones Actuales

El gobierno de los Países Bajos ha desarrollado en los últimos años, un visualizador de mapa digital de elevación de todo el país, el cual contiene datos de elevación detallados y precisos. Organizaciones dentro del país utilizan el sistema de visualización para la prevención frente a inundaciones. Los datos de puntos lo obtienen a través del escáner LiDAR, el cual tiene una precisión de 11 puntos por metro cuadrado. Una vez obtenidos los datos, hacen un procesamiento de clasificación según el tipo de terreno u objeto escaneado (agua, tierra, edificio, monumento, etc.) y también según el nivel de elevación, para finalmente poder mostrarlo en la plataforma web. (Actueel Hoogtebestand Nederland, 2020)

AHN Viewer

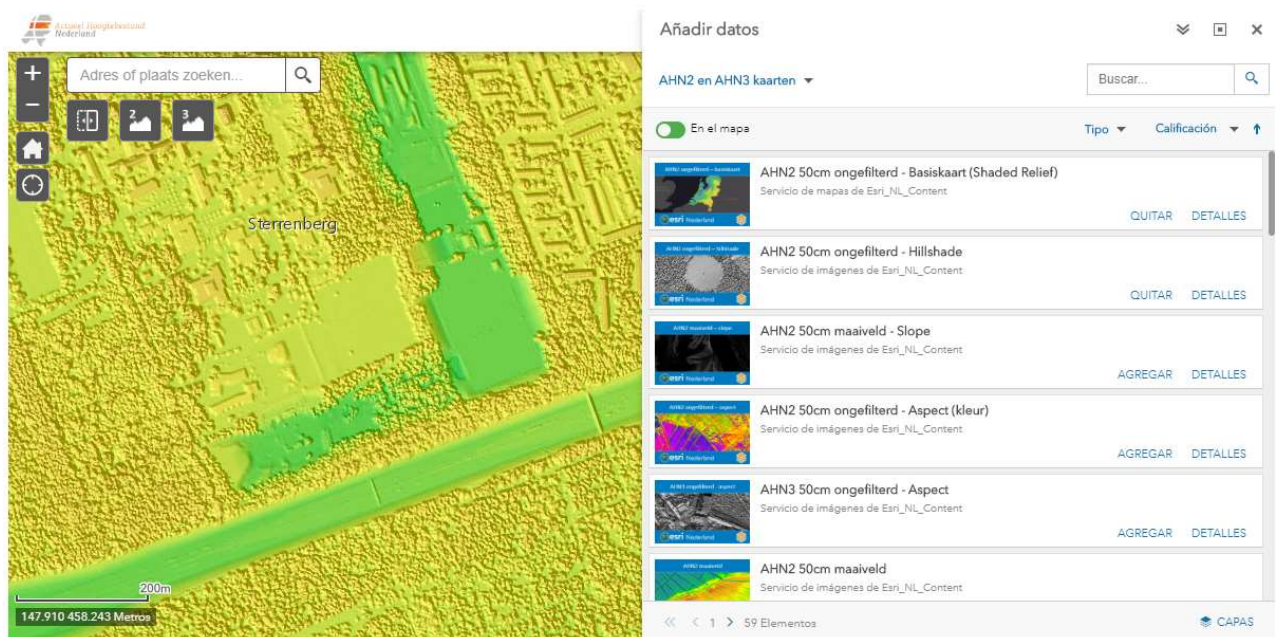


Ilustración 2.1 Imagen del sistema AHN Viewer y sus funcionalidades

3 Conclusiones

A partir de los artículos revisados para responder la primera pregunta de investigación, se puede concluir que la estructura de datos Octree es la adecuada para poder almacenar una gran cantidad de nube de puntos 3D y extraer su información de manera rápida y sin pérdidas para que puedan ser procesadas por los algoritmos de renderizado. Las características del Octree se adecuan bastante al tipo de trabajo que se está realizando, comprimen de manera significativa el tamaño de los archivos que contienen las nubes de puntos y su construcción no es muy costosa en tiempo para lo que ofrece. Definitivamente existen otras estructuras de datos que pueden cumplir las tareas de almacenamiento y extracción rápida, como el descrito por Schütz, Krösl y Wimmer; sin embargo, estas están destinadas para otro tipo de visualizaciones (realidad virtual) y tareas más complejas. (Schütz, Krösl & Wimmer, 2019)

Los artículos encontrados para responder la segunda pregunta de investigación presentan estudios sobre técnicas de renderización, los cuales son mejorados año tras año. El presente trabajo se inclina más a la técnica desarrollada por Schütz, el cual ofrece resultados de visualización de grandes entornos en alta definición y evita el aliasing. Además, ofrece una buena alternativa al iluminado, aplicando Eye-Dome-Lighting, añadiendo mayor realismo a la visualización sin ser tan costoso en las operaciones que realiza. Finalmente, debemos mencionar el trabajo realizado por Moreno, el cual hace hincapié en la optimización del algoritmo de renderizado que usa para reducir el costo en tiempo y procesamiento de los datos 3D, con el objetivo de que se pueda visualizar la mayor cantidad de objetos en una computadora o laptop con hardware de gama baja y/o media. (Moreno, 2014)

4 Referencias

Actueel Hoogtebestand Nederland (2020). AHN Viewer

Disponible en: <https://www.ahn.nl/ahn-viewer>

Bonatto, D., Rogge, S., Schenkel, A., Ercek, R., & Lafruit, G. (2016, December). Explorations for real-time point cloud rendering of natural scenes in virtual reality. In *2016 International Conference on 3D Imaging (IC3D)* (pp. 1-7). IEEE.

Cha, G., Park, S., & Oh, T. (2019). A terrestrial LiDAR-based detection of shape deformation for maintenance of bridge structures. *Journal of Construction Engineering and Management*, *145*(12), 04019075.

da Silva, V., Esperança, C., & Marroquim, R. (2019). OMiCroN—Oblique Multipass Hierarchy Creation while Navigating. *Computers & Graphics*, *84*, 42-54.

Discher, S., Richter, R., & Döllner, J. (2018, June). A scalable WebGL-based approach for visualizing massive 3D point clouds using semantics-dependent rendering techniques. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology* (pp. 1-9).

Elseberg, J., Borrmann, D., & Nüchter, A. (2011, October). Efficient processing of large 3d point clouds. In *2011 XXIII International Symposium on Information, Communication and Automation Technologies* (pp. 1-7). IEEE.

Elseberg, J., Borrmann, D., & Nüchter, A. (2013). One billion points in the cloud—an octree for efficient processing of 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, 76-88.

Evans, A., Agenjo, J., & Blat, J. (2014, November). Web-based visualisation of on-set point cloud data. In *Proceedings of the 11th European Conference on Visual Media Production* (pp. 1-8).

Gao, Z., Nocera, L., & Neumann, U. (2012, November). Visually-complete aerial LiDAR point cloud rendering. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 289-298).

Gong, J., Zhu, Q., Zhong, R., Zhang, Y., & Xie, X. (2012). An efficient point cloud management method based on a 3D R-tree. *Photogrammetric Engineering & Remote Sensing*, 78(4), 373-381.

INEI (2019). Documento Técnico y Normativo

Disponible en: <https://www.indeci.gob.pe/preparacion/documentos-tecnicos-y-normativos/>

Isa, S. M., Shukor, S. A., Rahim, N. A., Maarof, I., Yahya, Z. R., Zakaria, A., ... & Wong, R. (2018, December). A Review of Data Structure and Filtering in Handling 3D Big Point Cloud Data for Building Preservation. In *2018 IEEE Conference on Systems, Process and Control (ICSPC)* (pp. 141-146). IEEE.

Kang, L., Jiang, J., Wei, Y., & Xie, Y. (2019, June). Efficient Randomized Hierarchy Construction for Interactive Visualization of Large Scale Point Clouds. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)* (pp. 593-597). IEEE.

Kot, T., Novak, P., & Babjak, J. (2016, May). Visualization of point clouds built from 3D scanning in coal mines. In *2016 17th International Carpathian Control Conference (ICCC)* (pp. 372-377). IEEE.

Liu, Y., & Zhong, R. (2014, March). Buildings and terrain of urban area point cloud segmentation based on PCL. In *IOP Conference Series: Earth and Environmental Science* (Vol. 17).

Kharroubi, A., Hajji, R., Billen, R., & Poux, F. (2019). Classification and integration of massive 3d points clouds in a virtual reality (VR) environment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(W17), 165-171.

Ministerio de Agricultura (2017). Plan de Prevención ante la presencia de Fenómenos Naturales por Inundaciones, Deslizamientos, Huaycos y Sequías.
Disponibile en: <http://sigrid.cenepred.gob.pe/docs>

Moreno Valles, F. A. (2014). Diseño de un algoritmo para rendering eficiente de estructuras proteicas de gran escala.

Ponto, K., Tredinnick, R., & Casper, G. (2017, June). Simulating the experience of home environments. In *2017 International Conference on Virtual Rehabilitation (ICVR)* (pp. 1-9). IEEE.

Richter, R., & Döllner, J. (2010, June). Out-of-core real-time visualization of massive 3D point clouds. In *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (pp. 121-128).

Scheiblauer, C., & Wimmer, M. (2013). Analysis of Interactive Editing Operations for Out-of-Core Point-Cloud Hierarchies.

Sainz, M., & Pajarola, R. (2004). Point-based rendering techniques. *Computers & Graphics*, 28(6), 869-879

Sibbing, D., Sattler, T., Leibe, B., & Kobbelt, L. (2013, June). Sift-realistic rendering. In *2013 International Conference on 3D Vision-3DV 2013* (pp. 56-63). IEEE.

Schuetz, M., & Wimmer, M. (2018, August). Progressive real-time rendering of unprocessed point clouds. In *ACM SIGGRAPH 2018 Posters* (p. 41). ACM.

Schütz, M. (2016). Potree: Rendering large point clouds in web browsers. *Technische Universität Wien, Wien*.

Schütz, M., Krösl, K., & Wimmer, M. (2019, March). Real-time continuous level of detail rendering of point clouds. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 103-110). IEEE.

Venkateswaran, K., MacClune, K. and Enríquez, M.F. (2017). Learning from El Niño Costero 2017: Opportunities for Building Resilience in Peru/El Niño Costero: The Floods of 2017 in Peru.

Wand, M., Berner, A., Bokeloh, M., Fleck, A., Hoffmann, M., Jenke, P., ... & Schilling, A. (2007). Interactive Editing of Large Point Clouds. In *SPBG* (pp. 37-45).

You, Y., Fan, L., Roimela, K., & Mattila, V. V. (2014, September). Simple octree solution for multi-resolution lidar processing and visualisation. In *2014 IEEE International Conference on Computer and Information Technology* (pp. 220-225). IEEE.

